What is Claimed Is:

1. A method of externalizing legacy data from a legacy database on a data resource into a format compliant with a certain structured output format in which the format is specified by an associated, automatically generated meta-description, thus enabling access and processing of legacy data by applications that read the structured output format, the method comprising: an exploration/adaptation step for exploration of the data resource; and a production step for generation of the structured output format data out of the legacy data on the data resource.

2. A method of transforming legacy data from a legacy database on a data resource into Extensible Markup Language ("XML") format, the method comprising the steps of:

taking data modelled in a manner selected from a group consisting of relational data and data formatted according to an Abstract Syntax Notation ("ASN.1") data model and transforming such data into an XML-compliant data format, wherein an Extensible Markup Language meta-data description, contained in an XML Document Type Definition ("DTD"), is automatically generated.

3. The method of claim 2, wherein further are added an exploration/adaptation step for exploration of the data resource and a production step for generation of the Extensible Markup Language data out of the legacy data on the data resource.

4. A method of mapping data formatted according to either of a group consisting of a relational database model and an Abstract Syntax Notation ("ASN.1") data model into an Extensible Markup Language ("XML") compliant data format, the method including an exploration/adaptation step for exploration of legacy data in a legacy database on a data resource, and a production step for generation of XML data out of the legacy data, wherein the method performs data format mapping between the legacy database and an XML-compliant representation of that data.

5. The method of claim 4, wherein in the exploration step, standardized database functions are

used to retrieve information on a database's data scheme.

6. The method of claim 4 wherein the data format mapping is automated between legacy databases and an XML-compliant representation of that data.

7. The method of claim 6, wherein the mapping submethod covers all ASN.1 constructs, both primitive and composite.

8. The method of claim 2 wherein an XML Document Type Definition ("DTD") conversion between either of the group and an XML-compliant format is made.

9. A method that automates the data format mapping between data in a legacy database on a data resource formatted according to one member of a group selected from relational databases modeled according to a relational data model and Extensible Markup Language ("XML")-compliant representations of that data, wherein the method comprises:

(1) an exploration/adaptation step for exploration of the data resource, and

(2) a mapping step for generation of XML data out of the legacy data.

10. The method of claim 9 wherein the mapping step includes the following substeps:

(a) mapping relations, such as tables, within the relational data model onto XML elements including a "table" and a "row" element;

(b) mapping tuples, such as rows, within a relation onto XML elements which are nested within the XML "table" element; and

(c) mapping attributes (i.e., columns) of tuples onto XML elements which are nested within the XML "row" element.

11. The method of claim 4, 6, 7, or 9, wherein the mapping submethod takes ASN.1 constructs such as "ANY", "SEQUENCE", "SEQUENCE OF", "SET", "SET OF", "CHOICE", "COMPONENTS OF", "IMPORT", "TAG", and "EXPORT", and performs the following steps:

(a) mapping primitive ASN.1 types onto XML entities containing character data,

wherein, for each primitive ASN.1 type, one entity is defined;

(b) redefining fields of ASN.1 constructs, which are of a primitive ASN.1 type, as XML elements with an attribute of the corresponding entity;

(c) mapping the ASN.1 ANY type onto an XML entity containing uninterpreted data;

(d) mapping ASN.1 constants onto XML entities;

(e) mapping the ASN.1 SEQUENCE construct onto an XML element containing individual elements of the SEQUENCE as an XML sequence, wherein the XML "?" operator is applied to optional ASN.1 elements;

(f) mapping the ASN.1 SEQUENCE OF construct onto the XML repetition construct of elements "*";

(g) mapping the ASN.1 SET construct onto the XML CHOICE construct, wherein the XML "?" operator is applied to optional ASN.1 elements in a SET;

(h) mapping the ASN.1 SET OF construct onto the XML repetition construct of "*";

(i) mapping the ASN.1 CHOICE construct onto an XML element containing individual elements of the CHOICE as XML alternatives by using the "|" operator;

(j) mapping the ASN.1 COMPONENTS OF construct as follows:

for every COMPONENTS OF construct, creating an XML element which contains individual component elements, wherein the COMPONENTS OF optionally has an attribute that refers to the contained construct; and

(k) mapping each ASN.1 module having certain contents onto a specific XML Document Type Description ("DTD") by recursively applying the above rules to the module's contents, wherein each DTD forms a separate name space and the ASN.1 constructs IMPORT and EXPORT are then modeled using qualified names for imported/exported XML elements, thus creating an XML specification.

12. The method of claim 11, in order to perform a lossless mapping onto XML, further inserting ASN.1 tags into the XML specification by including one TAG element and three TAG attributes in the XML specification.

13. The method of claim 12, wherein the XML TAG elements use an attribute to indicate their

policy, their nature, and value of the tag and wherein, an additional attribute indicates whether IMPLICIT or EXPLICIT ASN.1 tagging has been used.

14. The method of claim 13 wherein additional attributes are included in the XML specification, wherein XML tag attributes in a separate name-space Abstract Syntax Notation are ASN:policy (EXPLICIT or IMPLICIT), ASN:class (UNIVERSAL, APPLICATION, CONTEXT, PRIVATE), and ASN:tag containing the original ASN.1 tag value.

15. A computer-readable medium encoded with a method of externalizing legacy data from a legacy database on a data resource into a format compliant with a certain structured output format in which the format is specified by an associated, automatically generated meta-description, thus enabling access and processing of legacy data by applications that read the structured output format.

16. The medium of claim 15, wherein further are added to the method an exploration/adaptation step for exploration of the data resource and a production step for generation of the structured output format data out of the legacy data on the data resource.

17. A computer-readable medium encoded with a method of transforming legacy data from a legacy database on a data resource into Extensible Markup Language ("XML") format, the method comprising the steps of:

taking data modelled in a manner selected from a group consisting of relational data and data formatted according to an Abstract Syntax Notation ("ASN.1") data model and

transforming such data into an XML-compliant data format, wherein an Extensible Markup Language meta-data description, contained in an XML Document Type Definition ("DTD"), is automatically generated.

18. The medium of claim 18, wherein further are added to the method an exploration/adaptation step for exploration of the data resource and a production step for generation of the Extensible

Markup Language data out of the legacy data on the data resource.

19. A computer-readable medium encoded with a method of mapping data formatted according to either of a group consisting of a relational database model and an Abstract Syntax Notation ("ASN.1") data model into an Extensible Markup Language ("XML") compliant data format, the method including an exploration/adaptation step for exploration of legacy data in a legacy database on a data resource, and a production step for generation of XML data out of the legacy data, wherein the method performs data format mapping between the legacy database and an XML-compliant representation of that data.

20. The medium of claim 19, wherein in the exploration step, standardized database functions are used to retrieve information on a database's data scheme.

21. The medium of claim 19 wherein the data format mapping is automated between legacy databases and an XML-compliant representation of that data.

22. The medium of claim 21, wherein the mapping submethod covers all ASN.1 constructs, both primitive and composite.

23. The medium of claim 17 wherein an XML Document Type Definition ("DTD") conversion between either of the group and an XML-compliant format is made.

24. A medium that automates the data format mapping between data in a legacy database on a data resource formatted according to one member of a group selected from relational databases modeled according to a relational data model and Extensible Markup Language ("XML")-compliant representations of that data, wherein the method comprises:

(a) an exploration/adaptation step for exploration of the data resource, and
(b) a mapping step for generation of XML data out of the legacy data.

25. The medium of claim 24 wherein the mapping step includes the following substeps:

(a) mapping relations, such as tables, within the relational data model onto XML elements including a "table" and a "row" element;

(b) mapping tuples, such as rows, within a relation onto XML elements which are nested within the XML "table" element; and

(c) mapping attributes (i.e., columns) of tuples onto XML elements which are nested within the XML "row" element.

26. The medium of claim 19, 21, 22, or 24, wherein the mapping submethod takes ASN.1 constructs such as "ANY", "SEQUENCE", "SEQUENCE OF", "SET", "SET OF", "CHOICE", "COMPONENTS OF", "IMPORT", "TAG", and "EXPORT", and performs the following steps:

(a) mapping primitive ASN.1 types onto XML entities containing character data, wherein, for each primitive ASN.1 type, one entity is defined;

(b) redefining fields of ASN.1 constructs, which are of a primitive ASN.1 type, as XML elements with an attribute of the corresponding entity;

(c) mapping the ASN.1 ANY type onto an XML entity containing uninterpreted data;

(d) mapping ASN.1 constants onto XML entities;

(e) mapping the ASN.1 SEQUENCE construct onto an XML element containing individual elements of the SEQUENCE as an XML sequence, wherein the XML "?" operator is applied to optional ASN.1 elements;

(f) mapping the ASN.1 SEQUENCE OF construct onto the XML repetition construct of elements "*";

(g) mapping the ASN.1 SET construct onto the XML CHOICE construct, wherein the XML "?" operator is applied to optional ASN.1 elements in a SET;

(h) mapping the ASN.1 SET OF construct onto the XML repetition construct of "*";

(i) mapping the ASN.1 CHOICE construct onto an XML element containing individual elements of the CHOICE as XML alternatives by using the "|" operator;

(j) mapping the ASN.1 COMPONENTS OF construct as follows:

for every COMPONENTS OF construct, creating an XML element which contains individual component elements, wherein the COMPONENTS OF optionally has an attribute that refers to the contained construct; and

(k) mapping each ASN.1 module having certain contents onto a specific XML Document Type Description ("DTD") by recursively applying the above rules to the module's contents, wherein each DTD forms a separate name space and the ASN.1 constructs IMPORT and EXPORT are then modeled using qualified names for imported/exported XML elements, thus creating an XML specification.

27. The medium of claim 26, in order to perform a lossless mapping onto XML, further inserting ASN.1 tags into the XML specification by including one TAG element and three TAG attributes in the XML specification.

28. The medium of claim 27, wherein the XML TAG elements use an attribute to indicate their policy, their nature, and value of the tag and wherein, an additional attribute indicates whether IMPLICIT or EXPLICIT ASN.1 tagging has been used.

29. The medium of claim 28 wherein additional attributes are included in the XML specification, wherein XML tag attributes in a separate name-space Abstract Syntax Notation are ASN:policy (EXPLICIT or IMPLICIT), ASN:class (UNIVERSAL, APPLICATION, CONTEXT, PRIVATE), and ASN:tag containing the original ASN.1 tag value.